

# Efficient Distance based Q-CSMA/CA Algorithms Wireless Multi Hop Networks

J.Joel,  
Student,  
PSN College of Engineering and Technology,  
Tirunelveli

T. Saravanan,  
Assistant Professor,  
PSN College of Engineering and Technology,  
Tirunelveli

## ABSTRACT

Carrier sense multiple access type algorithms results the maximum possible throughput in ad hoc networks. Even though the algorithm shows poor results in the delay performance also it shows some heuristic scheduling in the delay performance for the larger arrival rates. To overcome all these drawbacks we propose a time based CSMA algorithm which has the advantages of the followings, through this process multiple links are allowed to update their status in a single time period. Due to our proposed approach can achieve collision free transmission schedules during the control phase of the protocol. This algorithm shows the very good result in the delay performance for the throughput optimality property.

## 1. INTRODUCTION

For wireless networks with limited resources, resource allocation is an important part for a high performance and providing QoS. In our approach we find about the link scheduling for wireless networks, since the communication links in the network cannot transmit the data periodically as the interference occurs more. A scheduling process which determines which link can send or receive data at each time so that no interference occurs.

The parameters used to analyze the performance are the throughput and delay. The throughput of the scheduling algorithm will be determined by the mass of the arrival rate so that the scheduling algorithm makes the queue to maintain the stability of the process. The delay performance of the scheduling algorithm can be found out by the delay occur in the network. Since latest wireless networks require heavy bandwidths and delay requirements, designing such scheduling algorithms is important. We also in need of the scheduling algorithms to be distributed and having low complexity. There is no centralized approach and resource at every nodes in the nowadays wireless networks. Maximal scheduling is a low-complexity alternative to MWS, but it may only achieve a small fraction of the capacity region. Greedy Maximal Scheduling also known as Longest-Queue-First is another natural low-complexity alternative to MWS that has been observed to achieve good throughput and delay performance. GMS proceeds in a greedy manner by sequentially scheduling a link with the longest queue and disabling all its interfering links. It was shown that if the network satisfies the so-called local-pooling condition, then GMS is throughput-optimal. However, for networks with general topology, GMS may only achieve a fraction of the capacity region. Moreover, while the computational complexity of GMS is low, the signaling and time overhead of decentralization can increase with the size of the network.

There are several algorithms available for the CSMA, in the CSMA process a transmitter will have the task of sensing whether the channel is busy or not, before the data is transmitted. When the transmitter seems to identify whether

the network is busy, it waits for its turn (back off time). Since CSMA algorithms can be easily adoptable for the distributed manner, they are widely used in the protocols like IEEE802.11 and MAC protocols, etc., we propose a mathematical model which is used to calculate the throughput of CSMA protocols in multihop wireless networks.

One of our goals is to design distributed scheduling algorithms that have low complexity, are provably throughput-optimal, and have good delay performance. Toward this end, we design a discrete- time version of the CSMA-type random access algorithm that achieves the same product-form distribution over schedules. Our algorithm generates collision-free data transmission schedules while allowing for collisions during the control phase of the protocol thus relaxing the perfect CSMA assumption of the algorithms

## 2. COMMUNICATION OVERVIEW

We consider a wireless network with a graphical model  $G=(V,E)$  where  $V$  is the set of nodes and  $E$  is the links, nodes will be acting as both transmitter and receiver. In the wireless network for any link, the conflict link can represented  $C_{li}$  are the set of links where if any one of the links will be activated then the link cannot be established between the nodes, the following reasons are depicted for the deactivation for the links:

- The links that share a common node link, in this process, if two links share a common node cannot be activated simultaneously.
- Links may produce interference to links when transmitting: when the radio interference constrains when two links are close to each other cannot be activated.

A collision free graph  $G=(V,E)$  is the set of links that can be activated at the same time with respect to the conflict sets, when two links do not in a feasible schedule conflict with each other. Without misbehaving the rule of generality, we assume that all the links have unit capacity of transmitting one packet in one timeslot in a suitable scheduling condition.

As far as we know, the CSMA network model with exponential idle and transmission times was first considered in [8]. More recent work that models the backoff and transmission processes with exponential distributions. The assumption of exponential backoff time, however, is not compatible with practical CSMA protocols (e.g., 802.11), in which the backoff process has memory. Typically, the backoff process is controlled by a counter. When the counter is decremented to zero, then transmission begins. The countdown freezes whenever a neighbor starts to transmit. When the transmission of the neighbor completes, the countdown resumes with the previous counter value when the link was last frozen.

The scheduling is represented by the vector  $x \in \{0,1\}^{|E|}$ , the  $i$ th element of  $x$  is equal to 1 (ie  $x_i=1$ ).

$$x_i + x_j \leq 1, \quad (1)$$

A scheduling algorithm is the key way to decide which schedule to be used in every time of data transmission. We are focusing on MAC layer so we only consider the single hop traffic. The capacity region which is having the arrival rate  $\lambda$  where exist the scheduling algorithm which can maintain the queues the queues are bounded stochastic sense depending on the arrival model used. For the purposes we will assume that if the arrival process is stochastic, then the resulting queue length process admits a Markovian description, in which case stability refers to the positive recurrence of this Markov chain.

$$\Lambda = \{\lambda \mid E\mu \in C_0(M), \lambda < \mu\} \quad (2)$$

Where  $C_0(M)$  is the convex hull of the set of feasible schedules in  $M$ . When dealing with vectors, inequalities are interpreted component wise.

### 3. OPPURTUNISTIC SCHEDULING

Radio channel conditions vary independently for each user, in a given slot, there is a high probability of having a user whose channel state is near its peak. Scheduling such a node leads to high sum throughput. The gains are larger if the channel variations are larger which in turn are indeed larger if the number of nodes is large. Thus, the traditional view that rapid variations in the wireless channel pose a significant challenge for efficient communication has been converted into an opportunity for exploiting diversity.

We have a scheduling scheme where the scheduler picks up the node in in slot  $n$  such that

$$i_n = \arg \max_j x_n^j \quad (3)$$

This ‘pure’ opportunistic scheduling, though, maximizes overall sum throughput, is not necessarily fair. It may starve the users who have poor average channel states. We consider a wireless system where  $N$  users communicate with a base station. Nodes communicate with the base station using TDMA, i.e., time is divided into slots of equal duration and only one node can transmit in a slot. We assume that the slot duration is normalized to unity. The base station is the centralized entity that makes the scheduling decision and the user scheduled by the base station transmits in a slot. Exploiting multiuser diversity in an opportunistic manner by scheduling the node with the best channel state might introduce unfairness. Nodes that are closer to the base station might experience perennially better channel conditions and thereby obtain a higher share of the system resources at the expense of nodes that are farther away from the base station. On the other hand, scheduling nodes with poor channel states results in a reduction in the overall throughput.

Different scheduling algorithms provide fairness over different time intervals. A scheduling algorithm is long term fair if it provides a fair share of a certain quantity such as fraction of time slots or throughput to all nodes over a long period of time. As outlined earlier, the average throughput achieved by a node  $i$  over a long period of time can be expressed as:

$$\bar{T}^i = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M I_n^i R_n^i \quad (3)$$

On the other hand, a scheduling algorithm is short term fair if it provides a fair allocation of a certain quantity such as fraction of time slots or throughput to all users in an interval of  $M$  slots.

### 3.1 The Scheduling Algorithm is depicted below:

1. In the control slot, randomly select a decision schedule  $\mathbf{m}(t) \in \mathcal{M}_0$  with probability  $\alpha(\mathbf{m}(t))$ .  
 $\forall i \in \mathbf{m}(t)$ :  
 If no links in  $\mathcal{C}(i)$  were active in the previous data slot, i.e.,  $\sum_{j \in \mathcal{C}(i)} x_j(t-1) = 0$   
 (a)  $x_i(t) = 1$  with probability  $p_i$ ,  $0 < p_i < 1$ ;  
 (b)  $x_i(t) = 0$  with probability  $\bar{p}_i = 1 - p_i$ .  
 Else  
 (c)  $x_i(t) = 0$ .  
 $\forall i \notin \mathbf{m}(t)$ :  
 (d)  $x_i(t) = x_i(t-1)$ .
2. In the data slot, use  $\mathbf{x}(t)$  as the transmission schedule.

The transmission schedule used in the previous data slot and the decision schedule selected in the current control slot both are feasible, then the transmission schedule generated in the current data slot is also feasible.

### 4. HIGH PERFORMANCE AODV ALGORITHM

The AODV routing protocol is a reactive distance vector routing protocol that has been optimized for mobile ad-hoc wireless networks. AODV borrows basic route establishment and maintenance mechanisms from the Dynamic Source Routing (DSR) protocol, and hop-to hop routing vectors from the Destination-Sequenced Distance-Vector (DSDV) routing protocol. To avoid the problem of routing loops, AODV makes extensive use of sequence numbers in control packets.

When a source node needs to discover a route, it first executes the Route Optimization Function (ROF). The ROF first scans the existing routing tables and finds the interfaces which are not being used in any of the active data connections. In case none of the interfaces is free, it examines the Network Interface Queue (IFQ) of each interface. The IFQ is a drop-tail FIFO buffer, established between the Link and MAC layers, and holds packets which are to be transmitted on to the Physical Layer.

All interfaces, which have an IFQ length below a certain threshold, are selected as possible candidates for the Optimal Channel selection. The channel information of these free or least loaded interfaces is then retrieved through a Physical Layer hook. A random channel is then selected out of the free or least loaded channels, which is called the Optimal Channel. By randomizing the Optimal Channel, we maximize the channel distribution in two or more adjacent flows. The ROF returns this channel to the calling function.

The following figure depicts our proposed methodology:

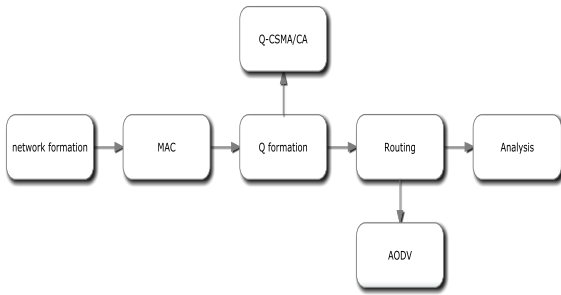


Fig 1 depicts the proposed architecture

Each intermediate node receiving the RREQ packet first verifies whether one of its interfaces is operating on the Recommended Channel. If this is the case, it creates the Reverse Route using this interface. In case the intermediate node has no interface operating on the Recommended Channel, it finds an Optimal Channel from the received RREQ packets and creates the Reverse Route on it. It then executes the ROF again to find the Optimal Channel to be recommended in the RREQ to the next hop neighbor. In case the intermediate node is a Mesh Router, it increments the Mesh Router Count. The intermediate node then broadcasts the RREQ on all interfaces.

#### 4.1 Throughput Optimality

Based on the product-form distribution, one can then proceed (under a timescale separation assumption to establish throughput optimality of the scheduling algorithm. Instead of pursuing such a proof here, we point out the intuition behind the proof of throughput optimality under the timescale separation assumption.

We associate each link  $i \in E$  with a nonnegative weight  $w_i(t)$  in timeslot  $x^*(t)$ . Recall that MWS selects a maximum-weight schedule in every timeslot such that

$$\sum_{i \in x^*(t)} w_i(t) = \max_{x \in M} \sum_{i \in x} w_i(t). \quad (4)$$

The probability of choosing a schedule is proportional to the exponent of its weight, so schedules with larger weight will be selected with higher probability. This is the intuition behind our proof. The same channel statistics tend to have the same average throughput and consequently the scheduling policy reduces to the opportunistic policy, i.e., in each slot, the user with the highest rate is scheduled. On the other hand, if the channel statistics of the users are not identical, then the users compete for resources based on their rates normalized by their respective throughputs. We find that the delay performance of Q-CSMA can be quite bad when the traffic intensity is high (this is also true in simulations of continuous-time CSMA algorithms) and much worse than GMS. However, GMS is a centralized algorithm and is not throughput-optimal in general. We are therefore motivated to design a distributed scheduling algorithm that can combine the advantages of both Q-CSMA (for achieving maximum throughput) and GMS (for achieving low delay). The same channel statistics tend to have the same average throughput and consequently the scheduling policy reduces to the opportunistic policy, i.e., in each slot, the user with the highest rate is scheduled. On the other hand, if the channel statistics of the users are not identical, then the users compete for resources based on their rates normalized by their respective throughputs.

## 5. EXPERIMENTAL RESULT

The experimental result can be carried out by the execution of the similar algorithm and taking the result of the algorithms and finding the optimality between them

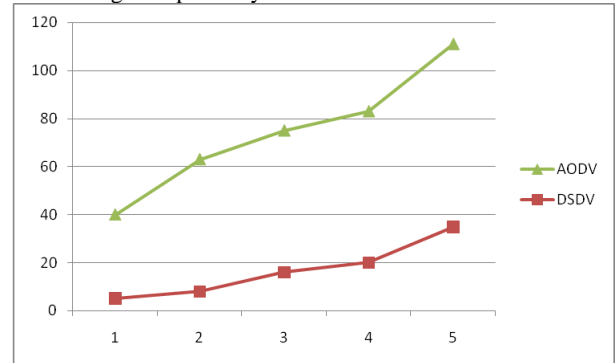


Fig 2 represents the comparison of the AODV algorithm

At low pause time (high mobility) the network topology will change frequently, more broken links will occur and the discovery process will be needed more. As a consequence, there will be a greater routing overhead and packets will be dropped resulting in more delay and less throughput.

The second graph depicts the throughput between the two algorithms

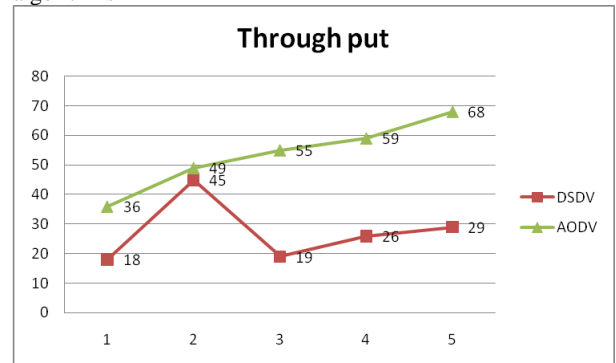


Fig 3 represents the comparison between the DSDV and AODV for throughput

This attribute can be explained by the fact that DSDV is a proactive routing protocol and in these types of protocols the path to a destination is immediately available. In other words, there is no delay caused by routing discovery. Furthermore, DSDV routing protocol tries to drop the packets, if it is not possible to deliver them which means less delay.

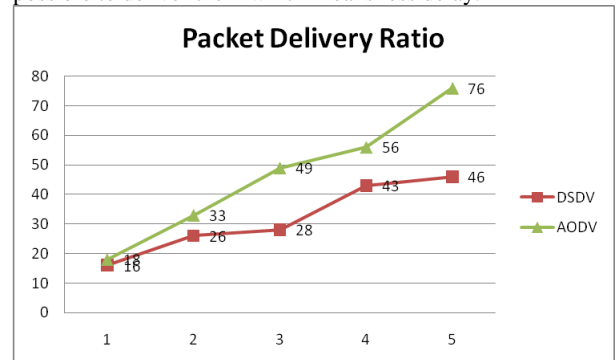


Fig 4 represents the Packet Delivery Ratio

The reason for the better packet delivery fraction in AODV over DSDV is that AODV protocol tries to guarantee that the packets will be delivered to the destination by delay

compromising. Whereas in DSDV, if it is not possible for the packets to be delivered, DSDV tries to drop them which means a lesser PDF as well as less delay. Furthermore, DSDV is a table-driven protocol and updates its table periodically which leads to an increase in the routing load in the network and less PDF. On the other hand, AODV is an on-demand routing protocol and adapts faster than DSDV to the change of the routing caused by mobile nodes in WSNs. However, PDF increases in both routing protocols in respect of pause time. This is because when the nodes are not moving too much the routing status becomes relatively stable and as a result a path finding process is not required.

## 6. CONCLUSION

AODV are simulated and compared under specific scenarios with WSNs environment. With the help of the NS2 simulator programme, DSDV and AODV are evaluated in respect of packet delivery fraction, end to end delay and average throughput. However, the simulation results reveal that there is no one protocol which is better than the other. Each protocol has its own advantages as well as its disadvantages making it suitable for some applications and not for others. Hence, an efficient routing protocol should be selected that suits the desired sensing task. Under packet delivery fraction, AODV has better performance than DSDV in the considered scenarios. PDF increases with an increase in the pause time for both protocols. As far as throughput is concerned, AODV performs by far better compared to DSDV. Average throughput in both protocols decreases steadily with an increase in the number of expired nodes and in case of pause time, the average throughput increases with increasing pause time. For the future we plan to prove such a conjecture for general networks as future work. To avoid the conjunction present in the network. And process on secure communication.

## 7. REFERENCES

- [1] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," in Proc. IEEE INFOCOM, April 2008, pp. 619–627.
- [2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in Proc. ACM SIGCOMM, 1994, pp. 212–225.
- [3] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. Sel. Areas Commun., vol. 18, no. 3, pp. 535–547, Mar. 2000.
- [4] R. R. Boorstyn, A. Kershnerbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet radio networks," IEEE Trans. Commun., vol. COM-35, no. 3, pp. 267–274, Mar. 1987.
- [5] C. Bordenave, D. McDonald, and A. Proutiere, "Performance of random medium access algorithms, an asymptotic approach," in Proc. ACM SIGMETRICS, Jun. 2008, pp. 1–12.
- [6] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in Proc. 43rd Annu. Allerton Conf. Commun., Control, Comput., Sep. 2005, pp. 28–30.
- [7] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," Adv. Appl. Probab., vol. 38, no. 2, pp. 505–521, 2006.
- [8] M. Durvy and P. Thiran, "A packing approach to compare slotted and non-slotted medium access control," in Proc. IEEE INFOCOM, Apr. 2006, pp. 1–12.
- [9] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," IEEE/ACM Trans. Netw., vol. 13, no. 2, pp. 411–424, Apr. 2005.
- [10] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," Found. Trends Netw., vol. 1, no. 1, pp. 1–149, 2006.
- [11] J. Ghaderi and R. Srikant, "On the design of efficient CSMA algorithms for wireless networks," Mar. 2010 [Online]. Available: <http://arxiv.org/abs/1003.1364>
- [12] L. Jiang and J. Walrand, "Convergence analysis of a distributed CSMA algorithm for maximal throughput in a general class of networks," UC Berkeley, Berkeley, CA, Tech. Rep., Dec. 2008.
- [13] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in Proc. 46th Annu. Allerton Conf. Commun., Control, Comput., Sep. 2008, pp. 1511–1519.
- [14] L. Jiang and J. Walrand, "Approaching throughput-optimality in a distributed CSMA algorithm with contention resolution," UC Berkeley, Berkeley, CA, Tech. Rep., March 2009.
- [15] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," IEEE/ACM Trans. Netw., vol. 17, no. 4, pp. 1132–1145, Aug. 2009.
- [16] F. Kelly, Reversibility and Stochastic Networks. Chichester, U.K.: Wiley, 1979.
- [17] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," IEEE/ACM Trans. Netw., vol. 19, no. 3, pp. 709–720, Jun. 2011.
- [18] S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-envelope computation of throughput distributions in CSMA wireless networks," IEEE Trans. Mobile Comput., vol. 9, no. 9, pp. 1319–1331, Sep. 2010.
- [19] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," IEEE J. Sel. Areas Commun., vol. 24, no. 8, pp. 1452–1463, Aug. 2006.
- [20] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Maximizing utility via random access without message passing," Microsoft Research, Tech. Rep., Sep. 2008.
- [21] P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable rate region of CSMA schedulers in wireless networks with primary interference constraints," in Proc. IEEE CDC, Dec. 2007, pp. 1156–1161.
- [22] J. Ni, B. R. Tan, and R. Srikant, "Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.
- [23] A. Proutiere, Y. Yi, and M. Chiang, "Throughput of random access without message passing," in Proc. CISS, Mar. 2008, pp. 509–514.

- [24] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in Proc. CISS, Mar. 2008, pp. 498–502.
- [25] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in Proc. ACM SIGMETRICS, Jun. 2009, pp. 133–144.
- [26] S. Shakkottai and R. Srikant, "Network optimization and control," *Found. Trends Netw.*, vol. 2, no. 3, pp. 271–379, 2007.
- [27] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximal throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
- [28] E. Vigoda, "A note on the Glauber dynamics for sampling independent sets," *J. Combin.*, vol. 8, pp. 1–8, 2001.
- [29] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in Proc. IEEE INFOCOM, Mar. 2005, vol. 1, pp. 23–34.
- [30] A. Warriier, S. Janakiraman, and I. Rhee, "DiffQ: Practical differential backlog congestion control for wireless networks," in Proc. IEEE INFOCOM, Apr. 2009, pp. 262–270.